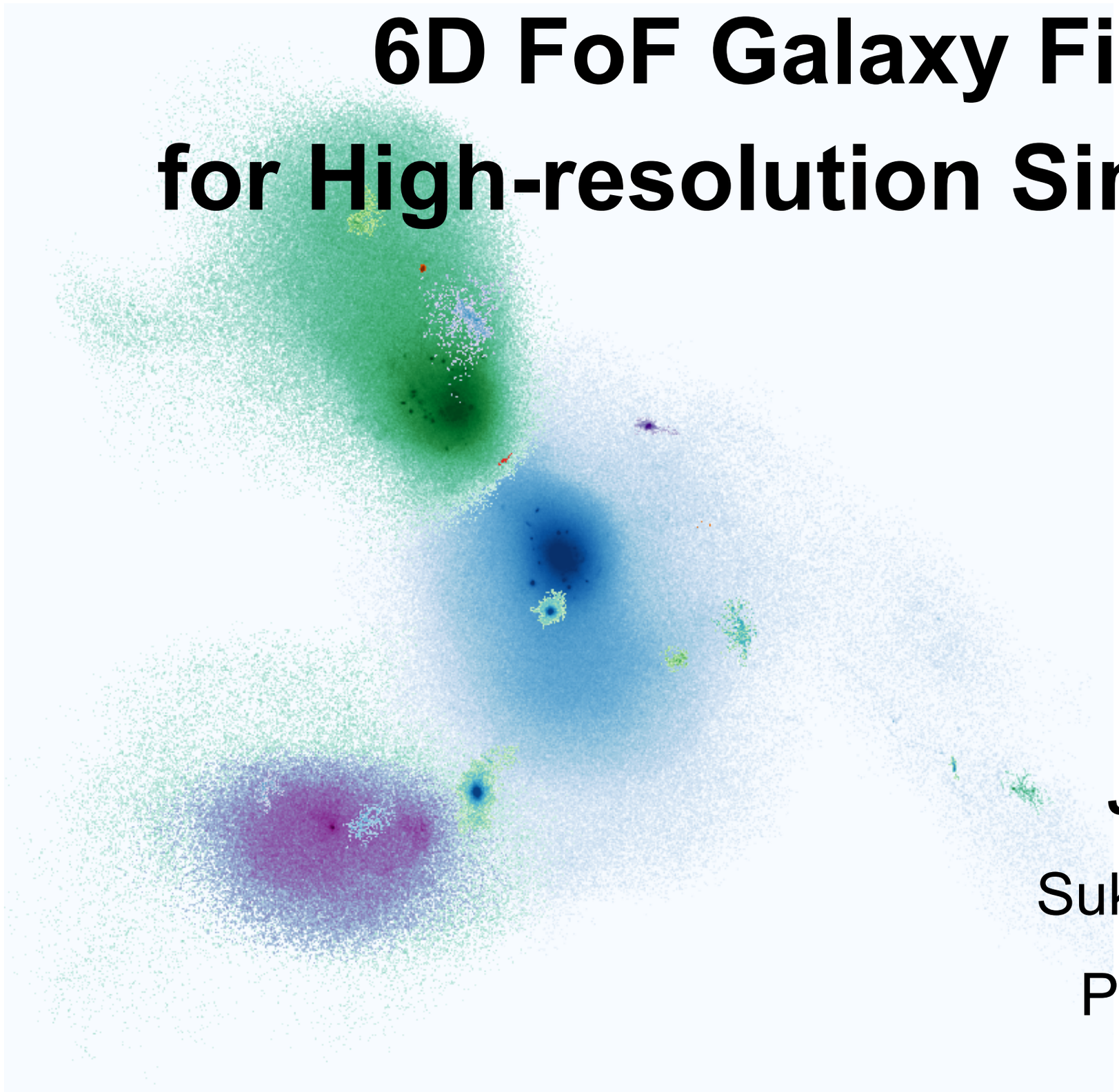
Performance Enhancement of 6D FoF Galaxy Finder for High-resolution Simulations



Jinsu Rhee (Yonsei)

Sukyoung K. Yi (Yonsei)

Pascal Elahi (Pawsey)

Friend-of-Friends Galaxy Identification

Why “FoF-based Galaxy Finder”

- ***Expandability to the phase-space***

  Definition of 6D metric is enough

  More reliable substructure identification in dense regions

- ***Shape-free: Versatile***

  From virialized objects to tidal streams

- ***Yet, Still Challenging***

  Sometimes, cost-inefficient identifications on high-resolution simulations

  (Behroozi et al. 2013; Feng & Modi 2017)

“6D Galaxy Identification on High-res Sim.”

VELOCIraptor-STF on NewHorizon

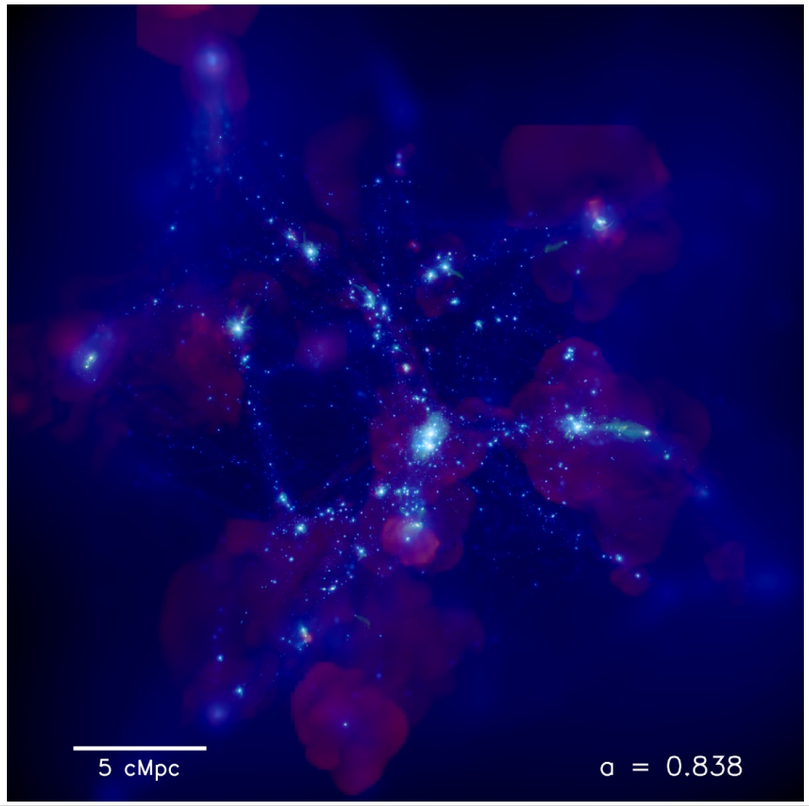
VELOCIraptor-STF *(Elahi+19)*

- 3D & 6D FoF Halo/Galaxy Finder
- Able to read various input types

  RAMSES, GADGET, …
- Available using multiple particle types

  DM / DM + Star / DM + Star + Gas

NewHorizon Simulation *(Dubois+21)*

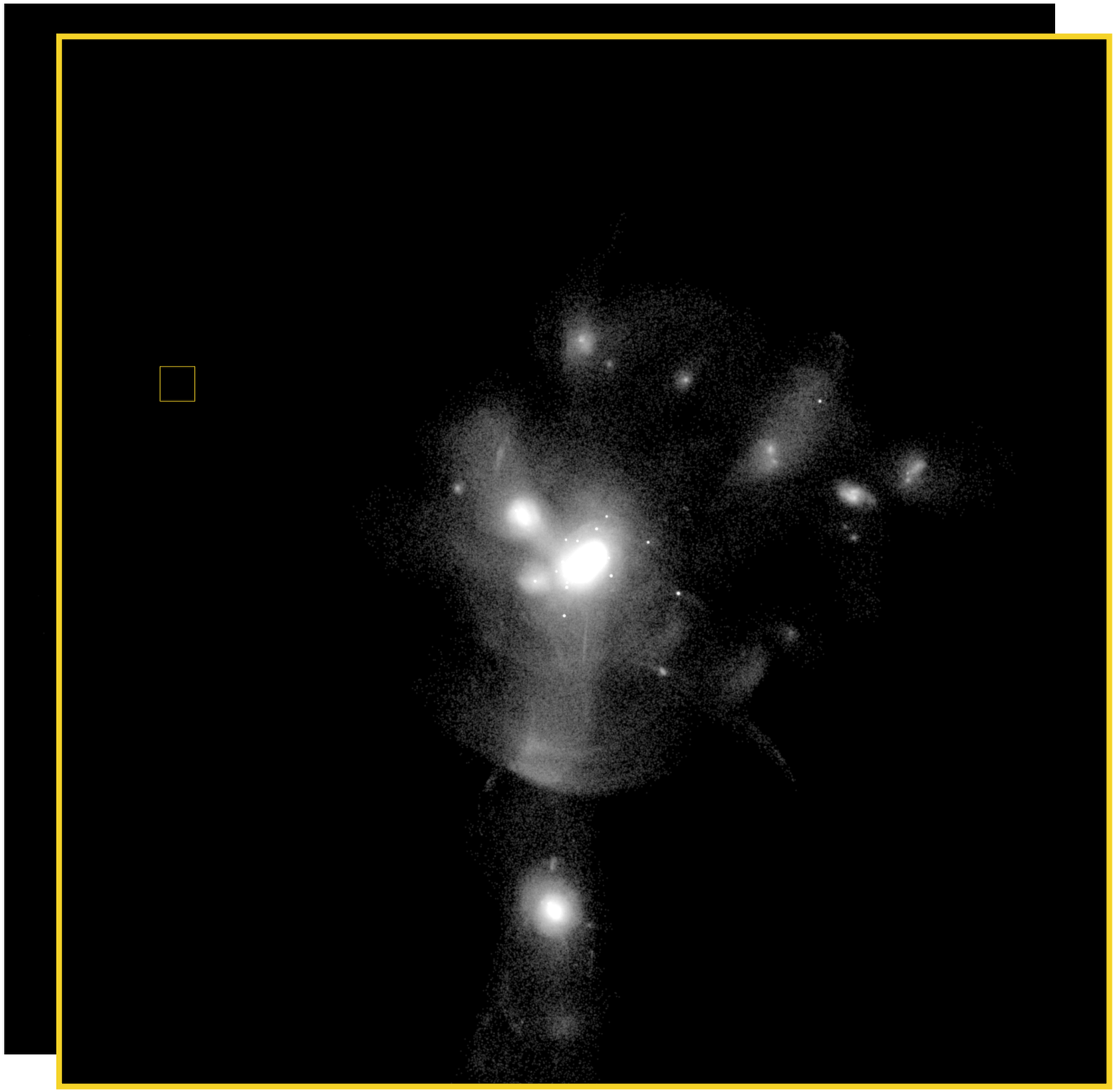


- Unreached high resolutions

  $\Delta X = 34\,\mathrm{pc};\quad M_{\mathrm{star}} \sim 10^4\,M_{\odot};$
- A good sample to test FoF galaxy finding

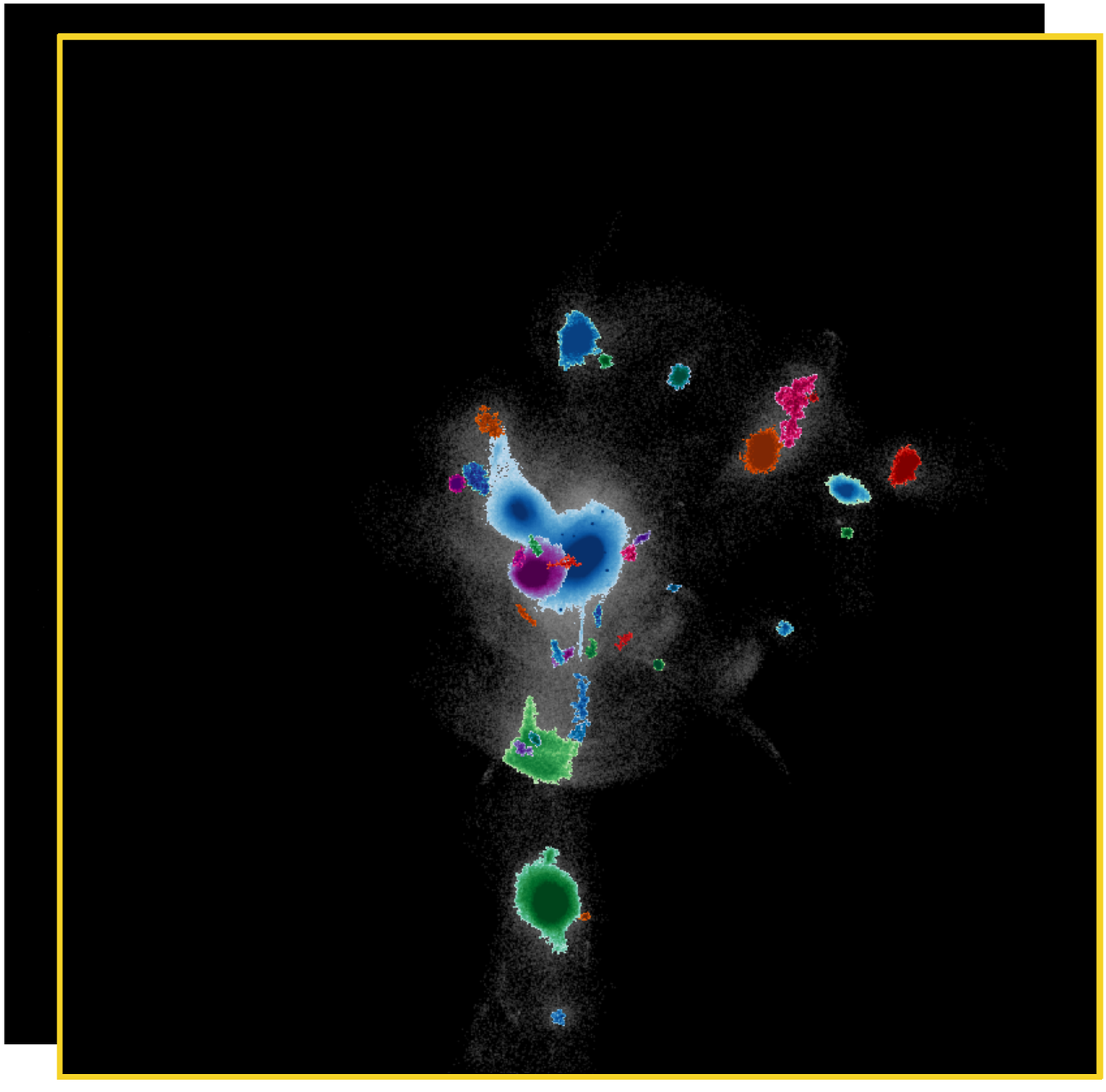
  on the next generation high-res simulations

Steps for Galaxy Identification



3D FoF

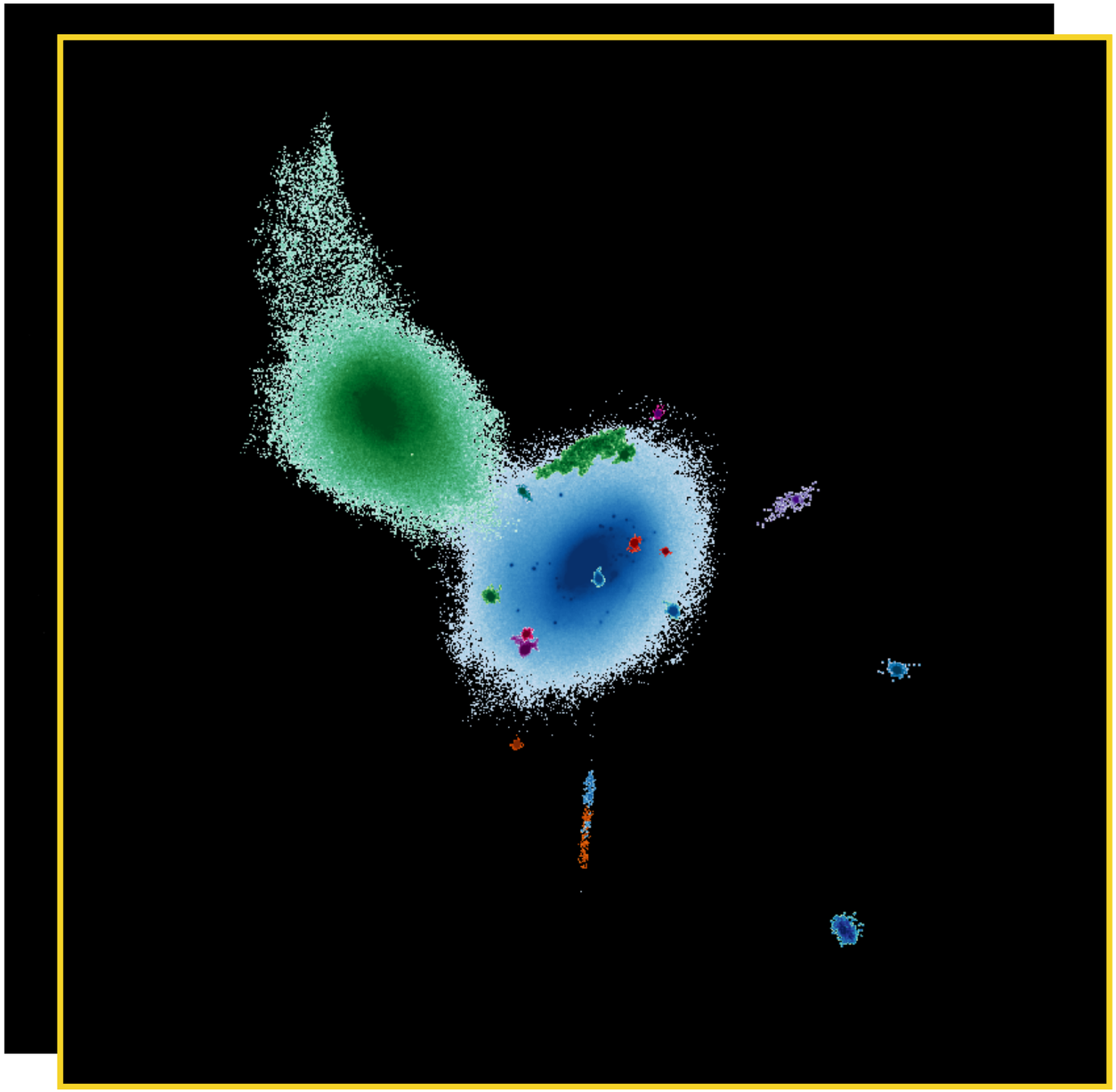
Steps for Galaxy Identification



3D FoF

6D FoF

Steps for Galaxy Identification

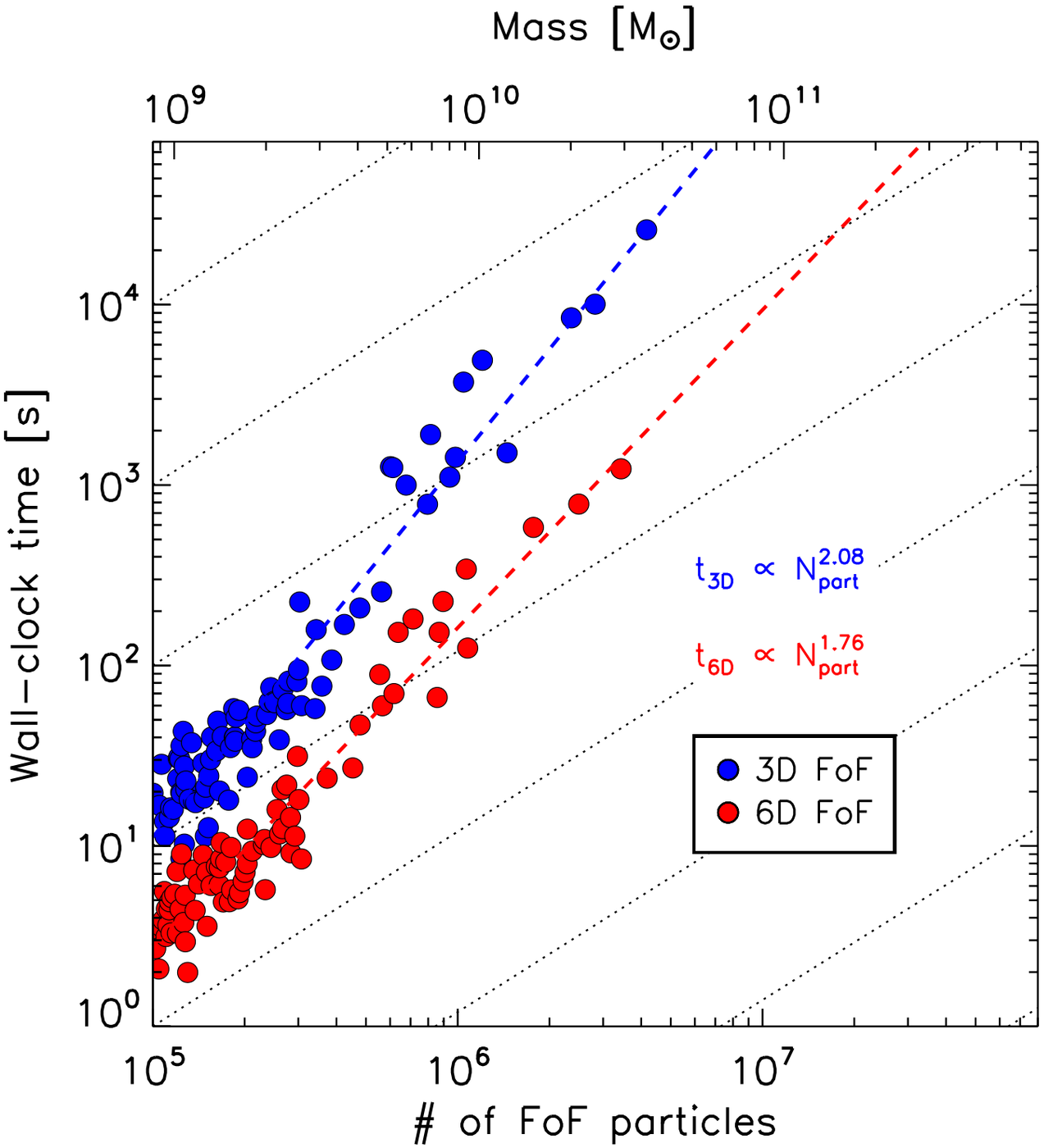


3D FoF

6D FoF

Galaxies

Poor Performance on NewHorizon



Slow Finding on NH

- **Snapshot of z = 2.46**

  N_{star} = 47,218,669

- **Very Slow**

  >10,000 sec (3D); >1,000 sec (6D)

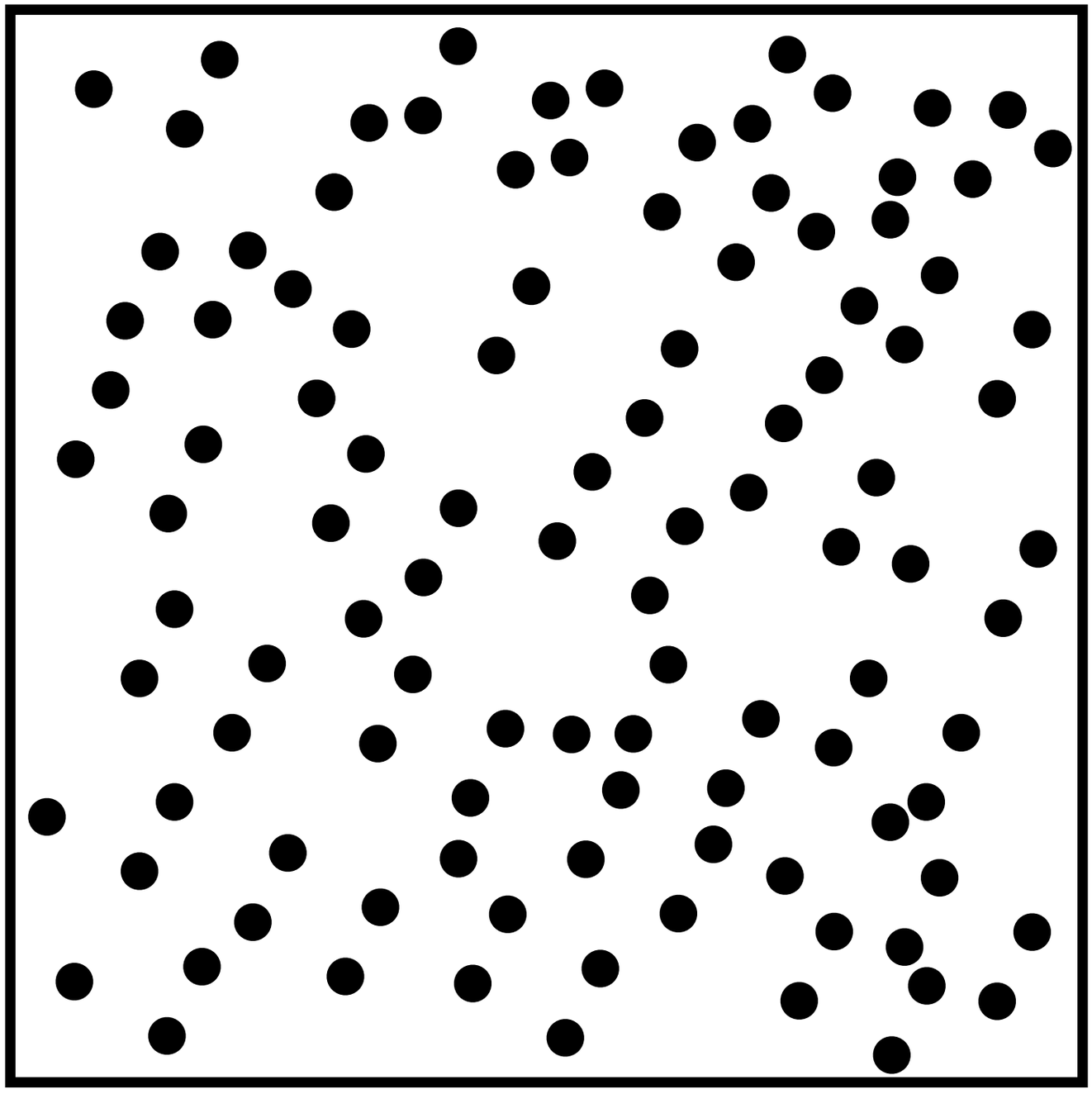
- **Poor Scaling**

  Impractically slow for more massive galaxies

- **Due to too many distance calculations at the central region**

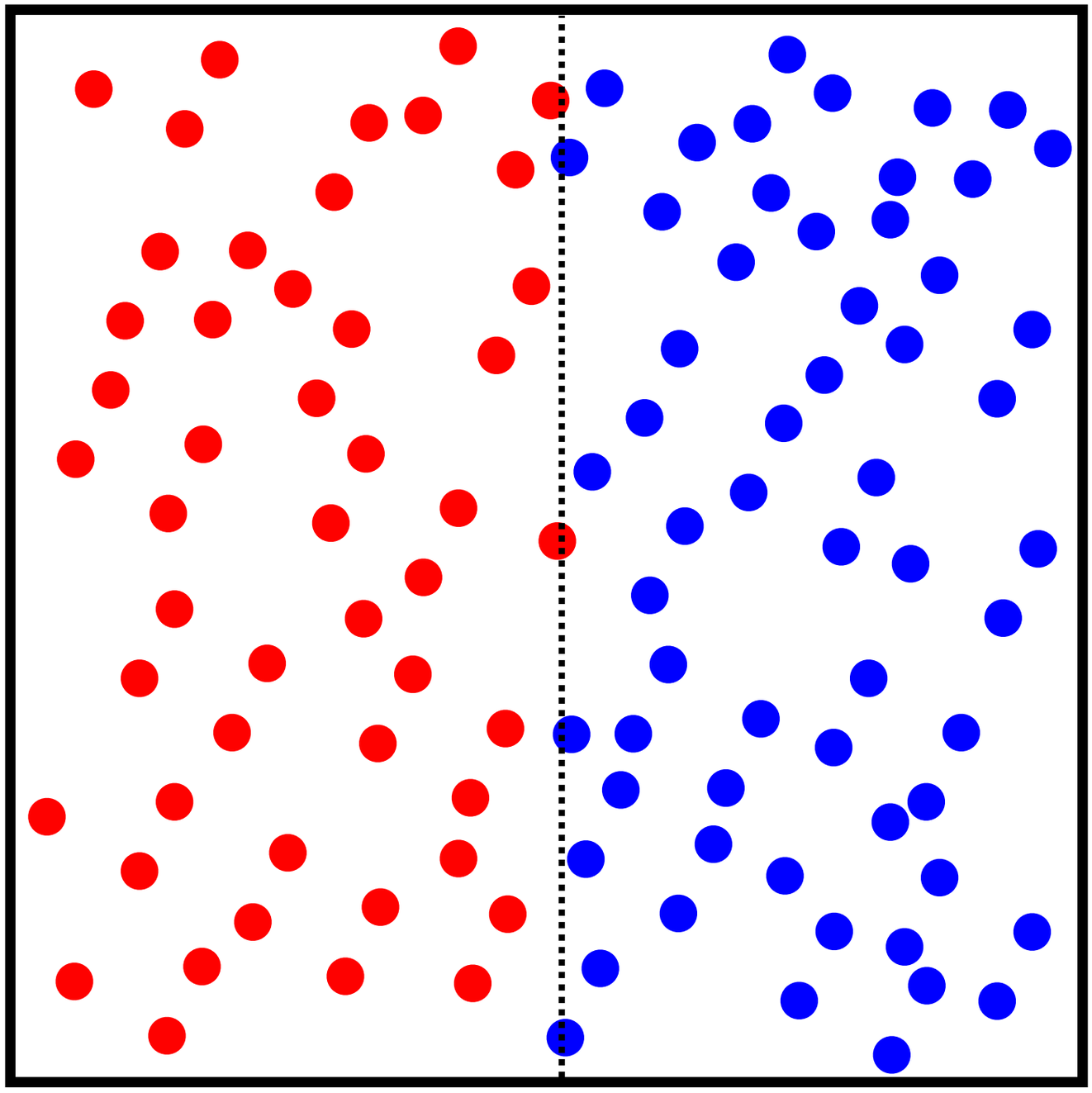
KD Tree for the Neighbor Query

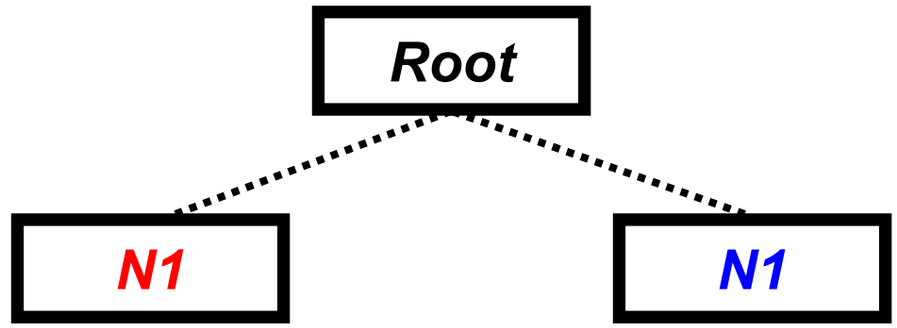
Distance calculation based on KD Tree



KD Tree for the Neighbor Query

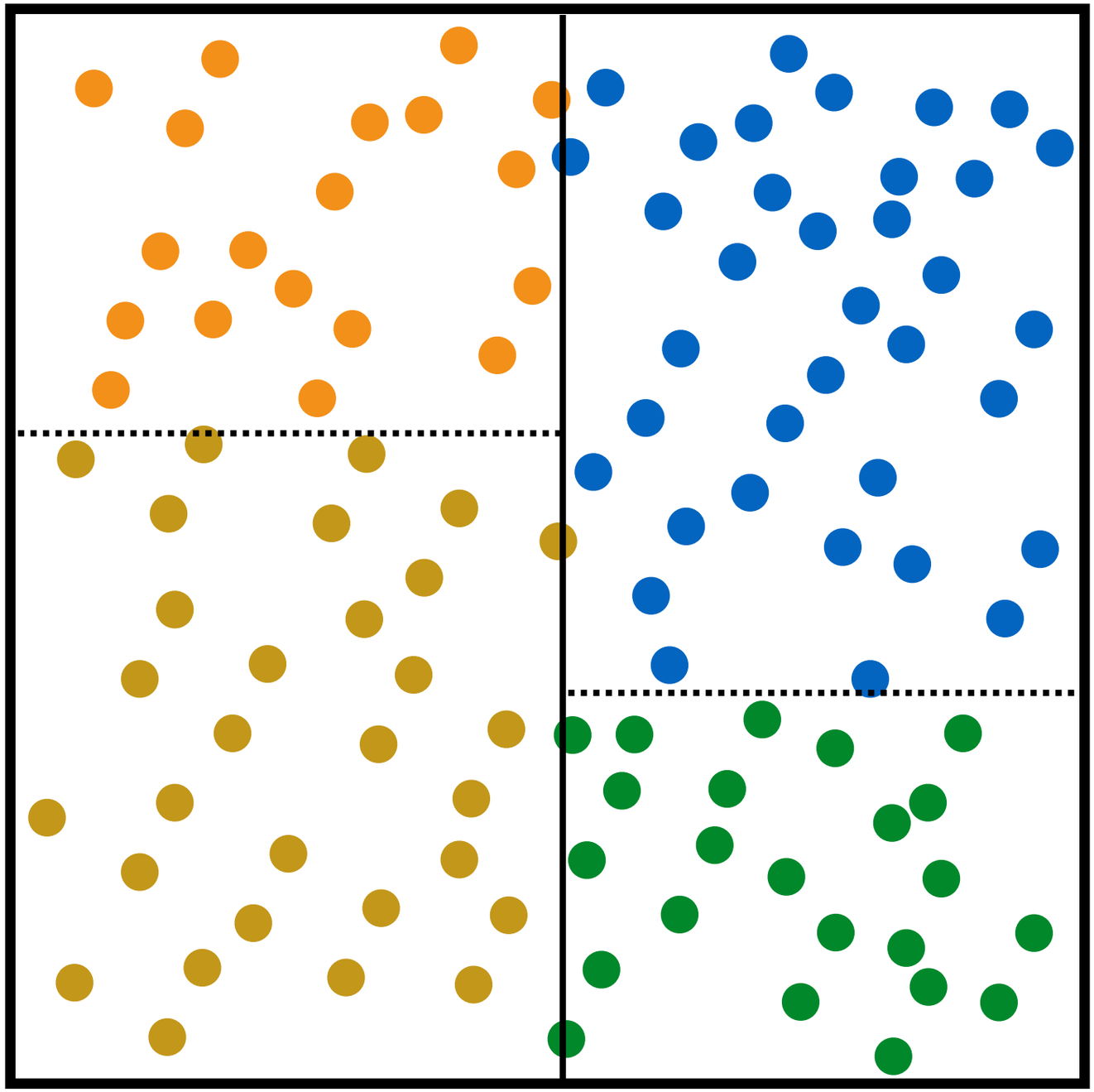
Distance calculation based on KD Tree

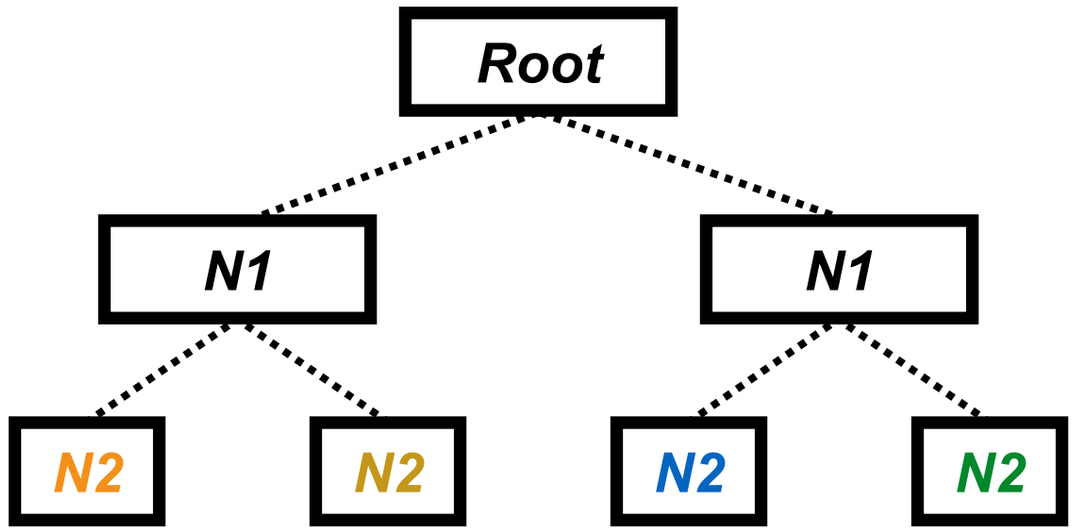




KD Tree for the Neighbor Query

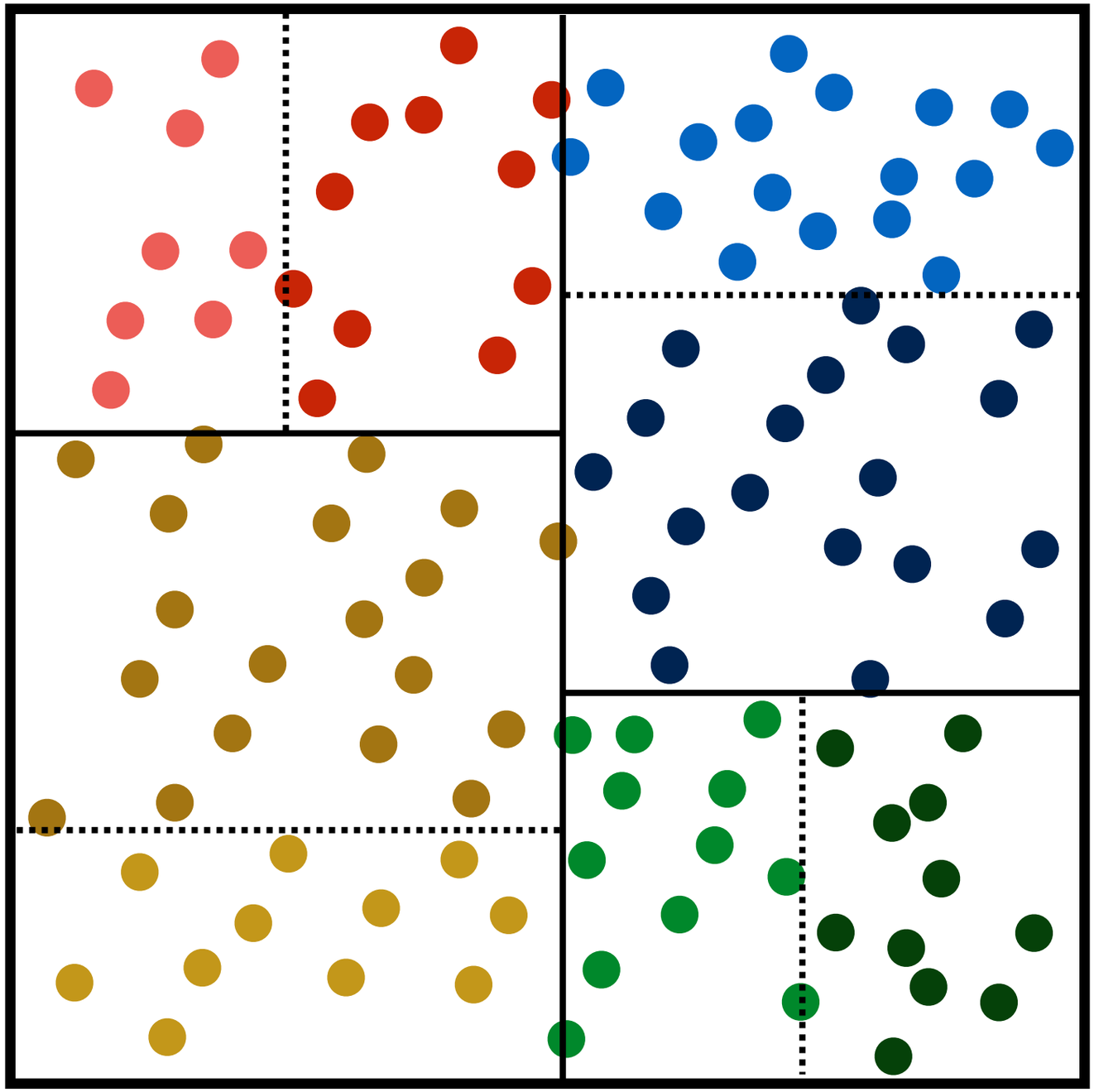
Distance calculation based on KD Tree

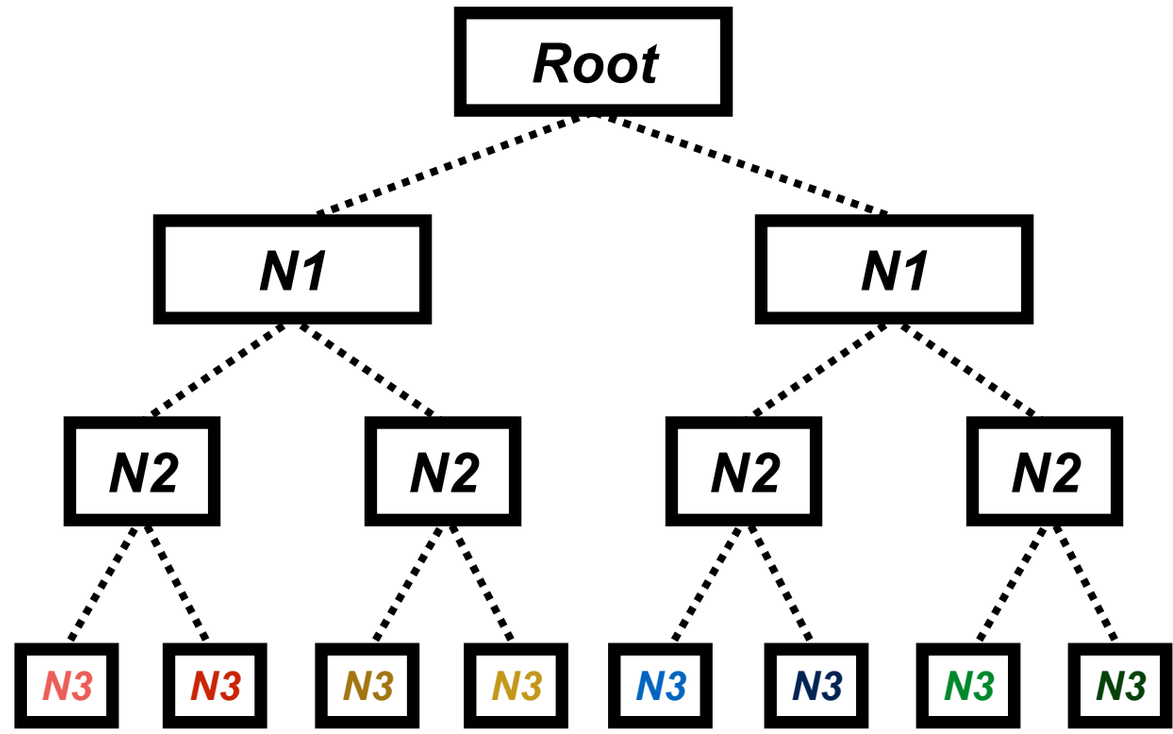




KD Tree for the Neighbor Query

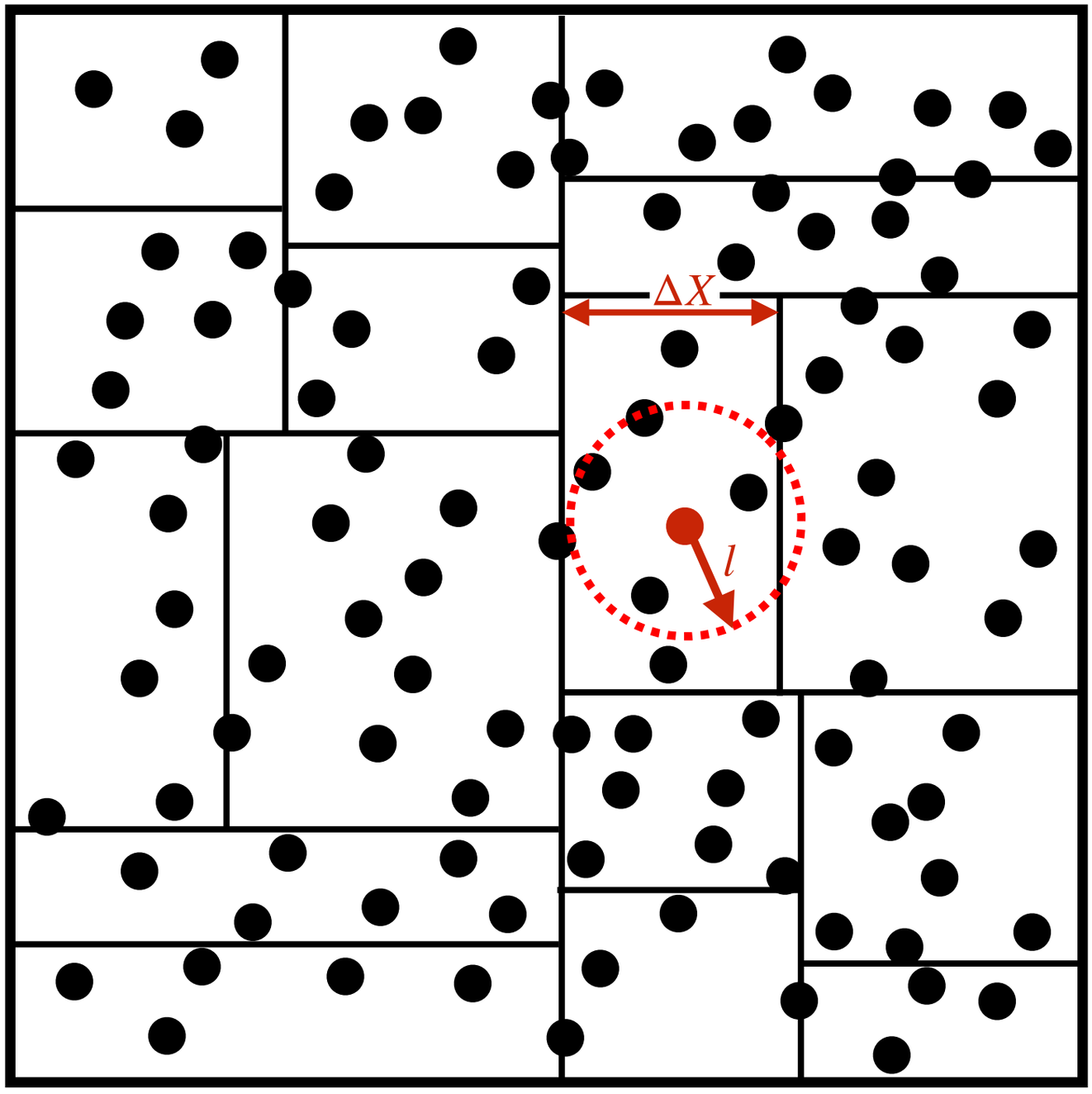
Distance calculation based on KD Tree





KD Tree for the Neighbor Query

Distance calculation based on KD Tree



To find particles $d < l$

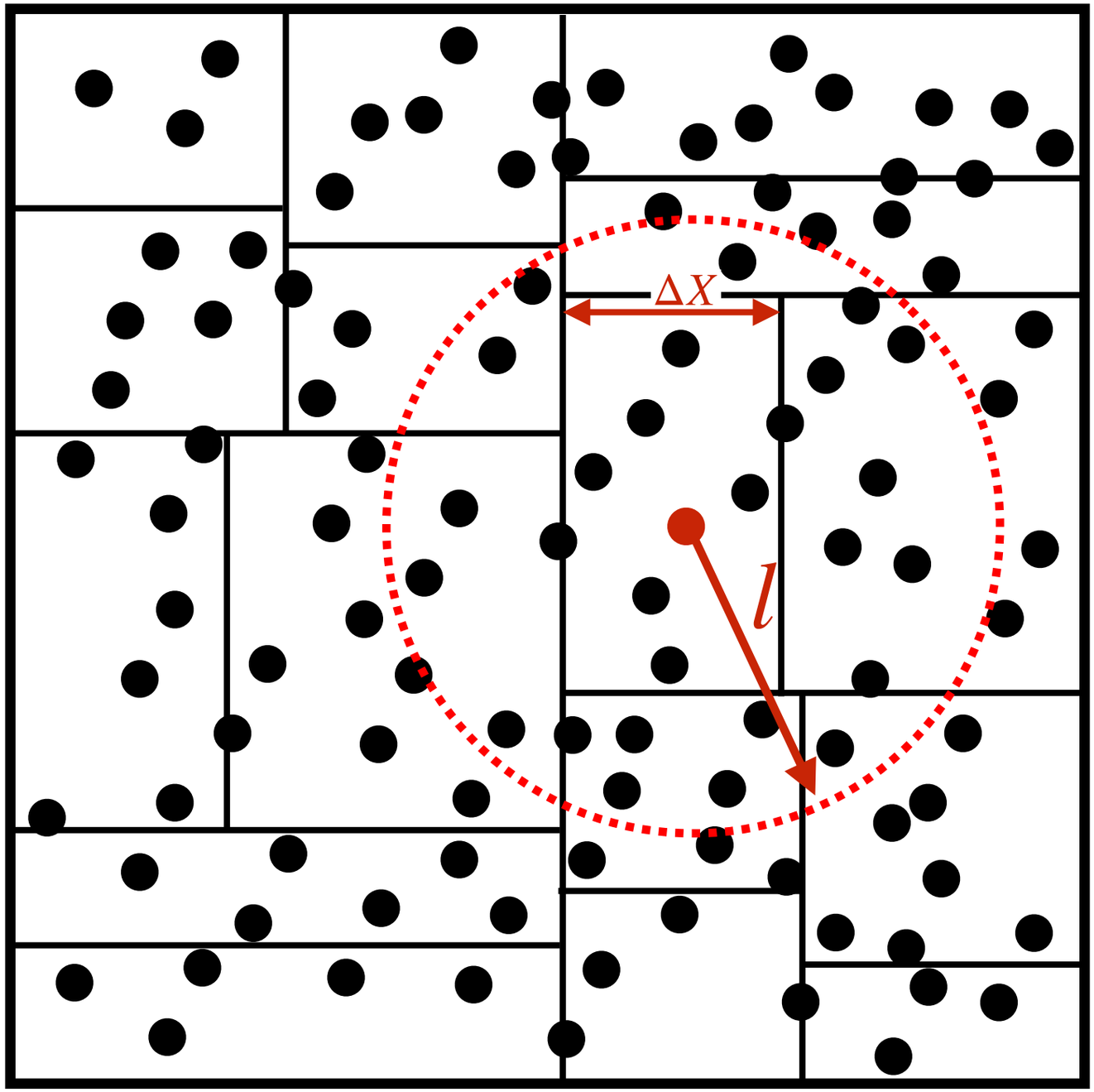
- Vanilla - $O(N^2)$
- KD Tree

  Usual case ($l \sim \Delta X$)

  - $O(N \log N)$ few nodes included

KD Tree for the Neighbor Query

Distance calculation based on KD Tree



To find particles $d < l$

- Vanilla - $O(N^2)$
- KD Tree

  Usual case ($l \sim \Delta X$)

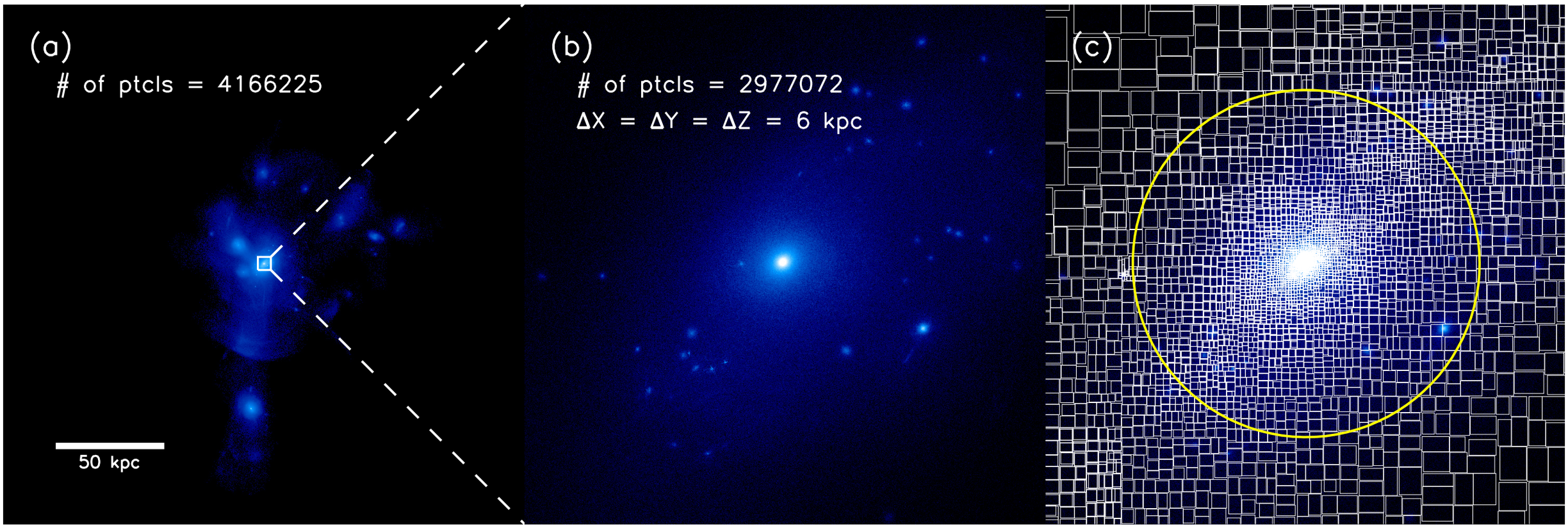
  - $O(N \log N)$

  High-density case ($l \gg \Delta X$)

  - $O(N_{\text{core}}^2)$ many nodes included

The Origin of the Slow Performance

NH is such a case



High resolutions of NH allows the very high central number density ($n_{\rm core} = 10^6\,{\rm kpc}^{-3}$)

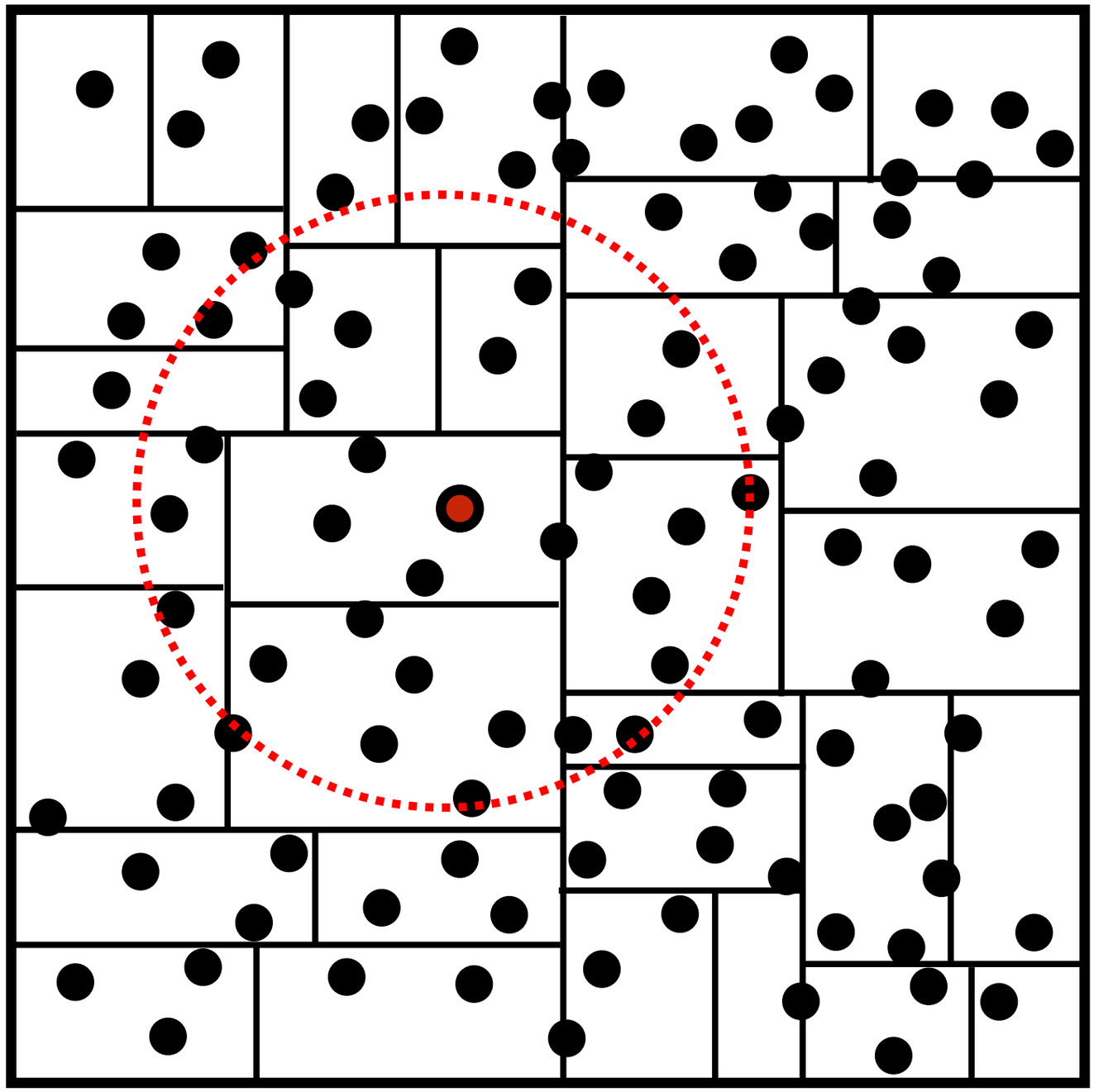
- # of particles within a linking length = 1,600,000
- # of nodes within a linking length = 72,087

Implementations

Strategy

- Node-Closure

  Close node if its all particles are linked

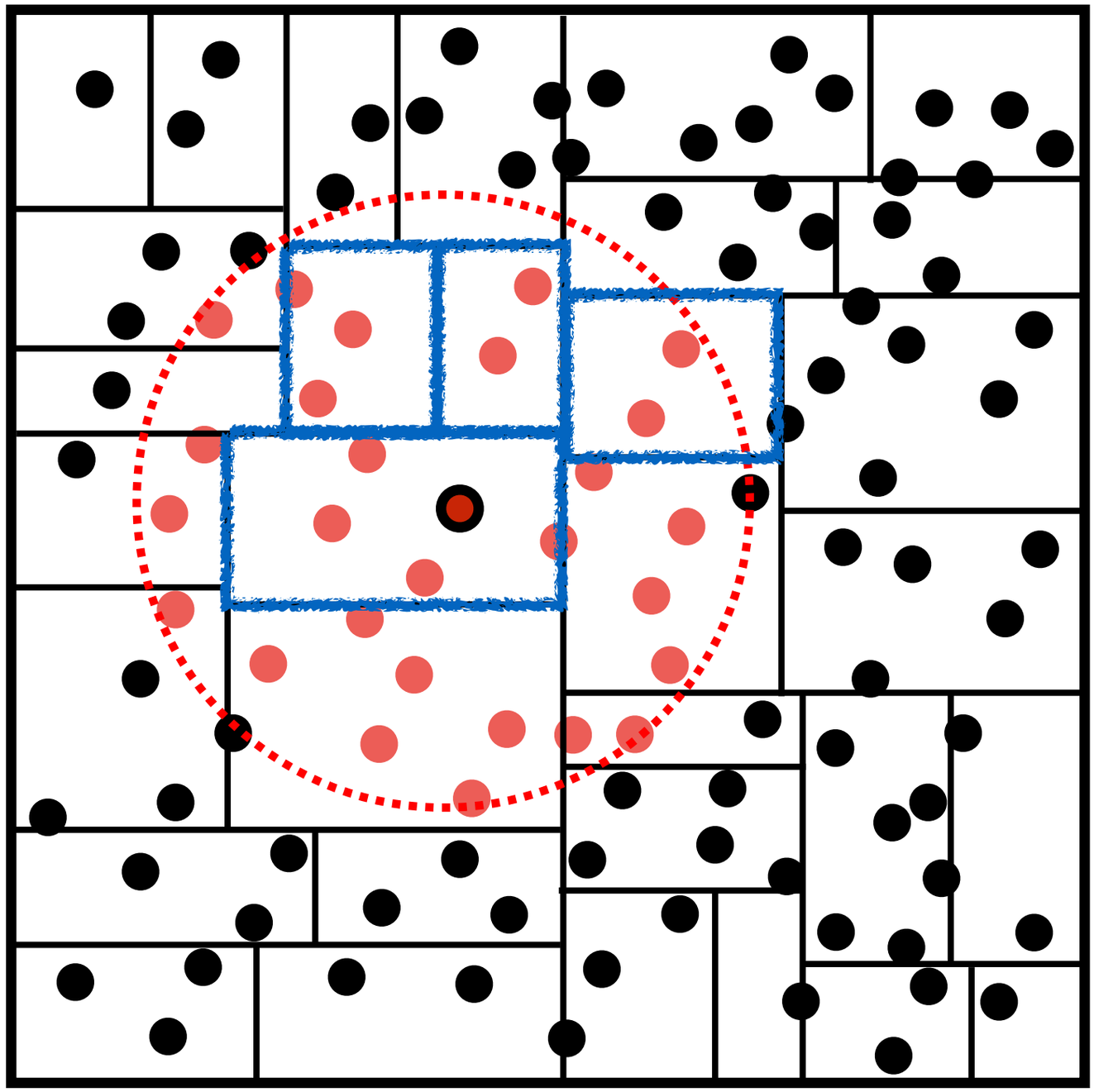


Implementations

Strategy

- Node-Closure

  Close node if its all particles are linked

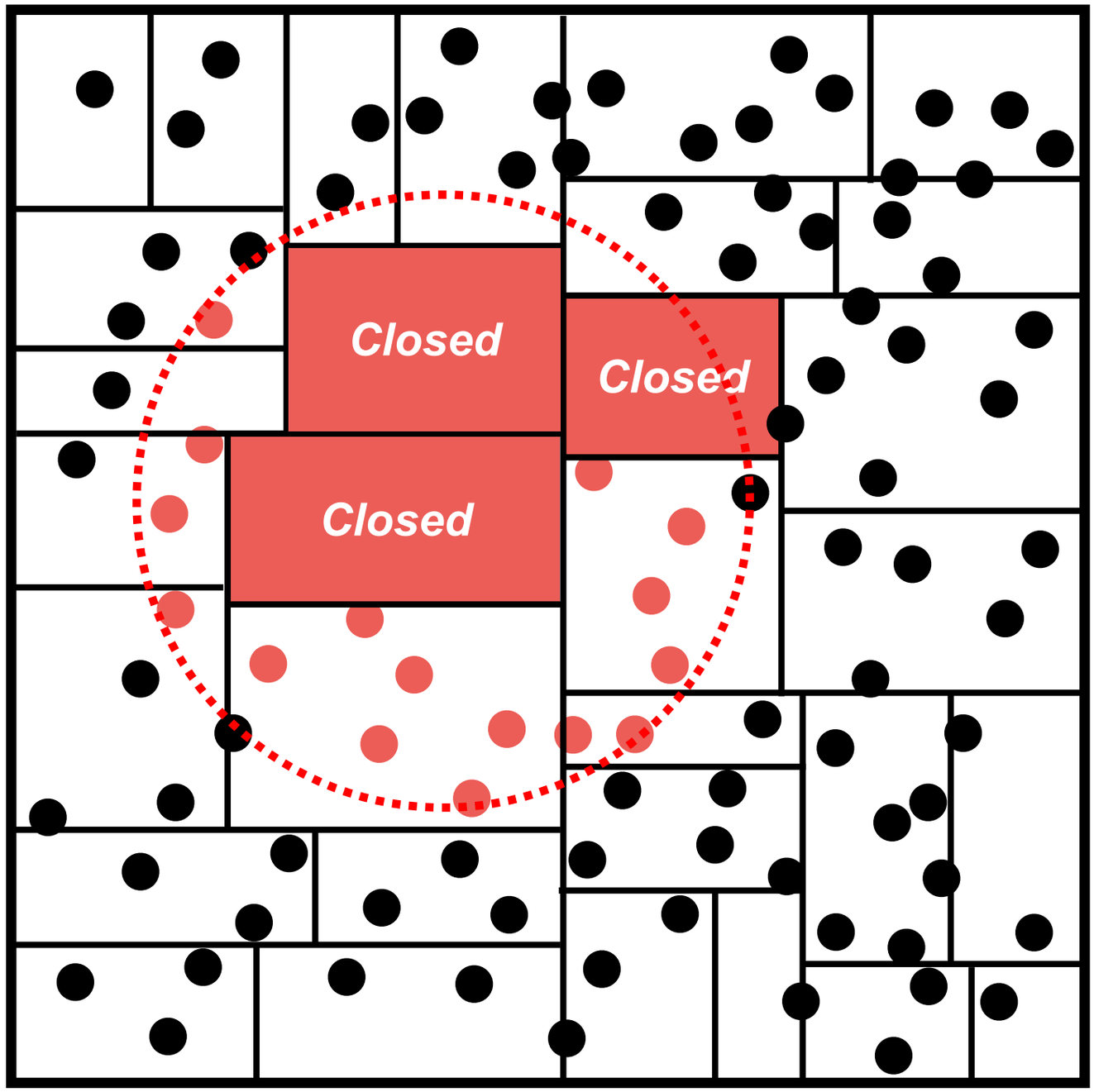


Implementations

Strategy

- Node-Closure

  Close node if its all particles are linked

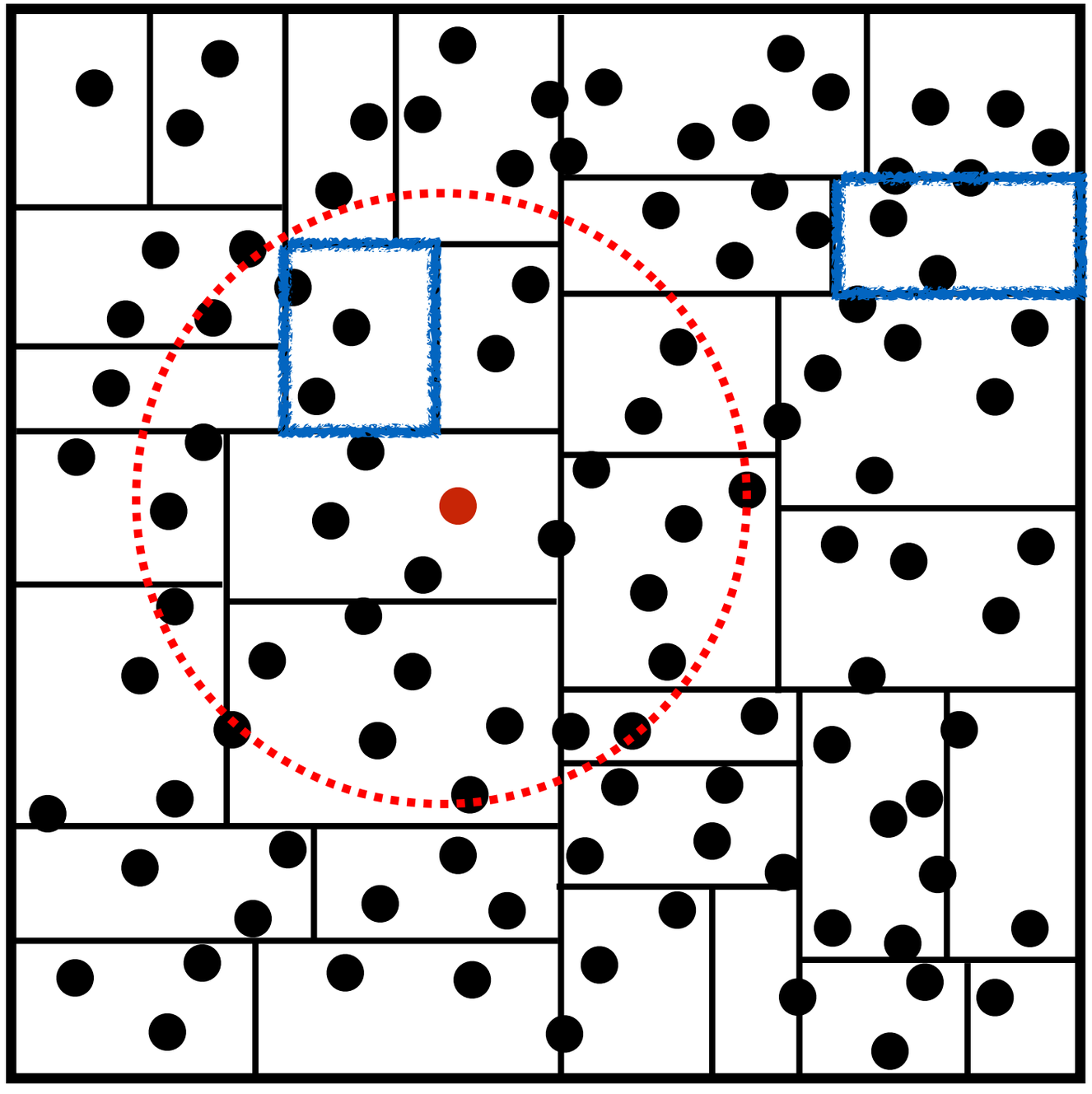


Implementations

Strategy

- Node-Closure
- Node-Geometry

  If a node is fully outside or inside



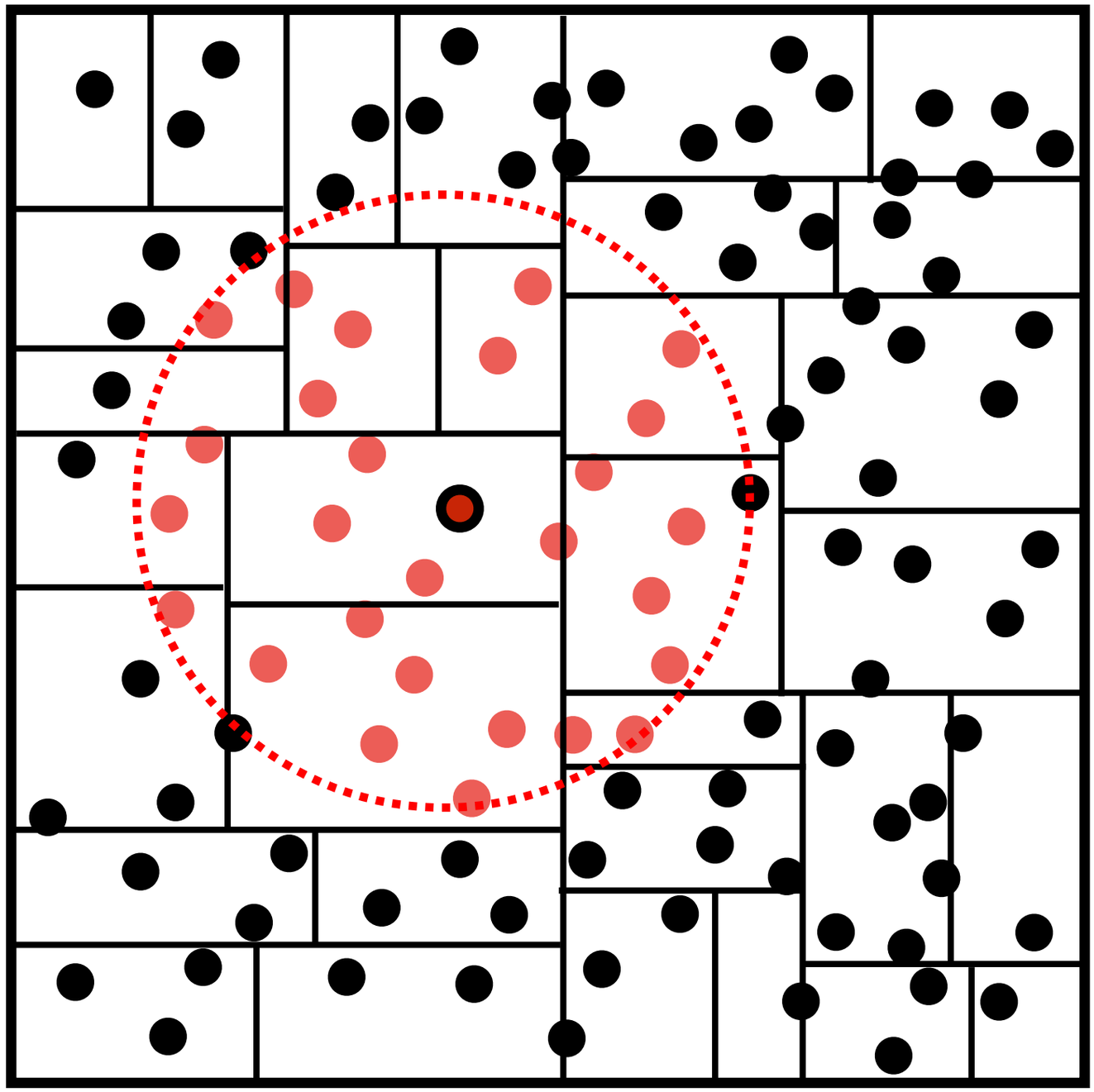
Implementations

Strategy

- Node-Closure
- Node-Geometry
- Splay

  Effectively find new member particles.

  But, do not affect the number of calculations.



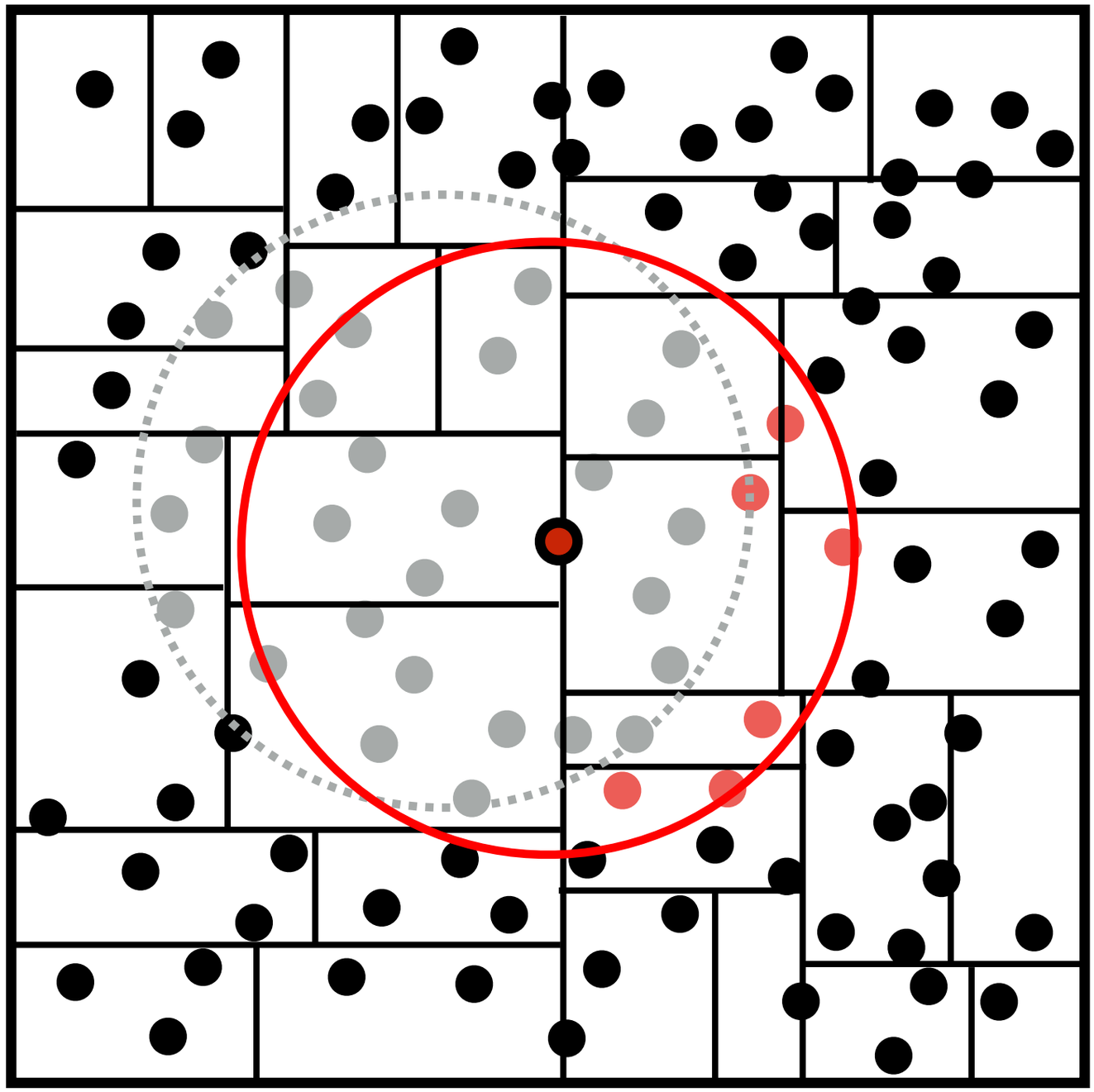
Implementations

Strategy

- Node-Closure
- Node-Geometry
- Splay

  Effectively find new member particles.

  But, do not affect the number of calculations.



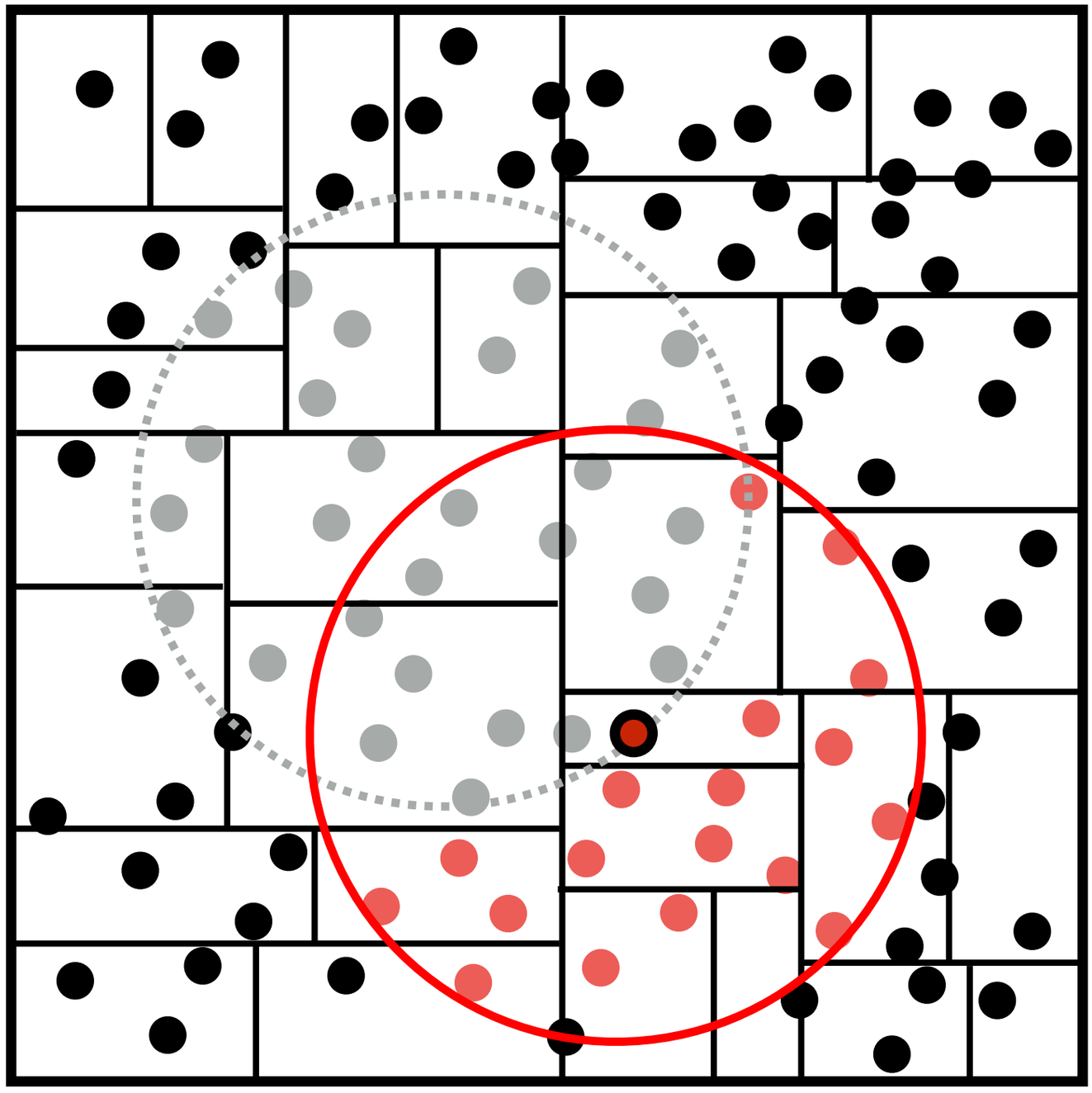
Implementations

Strategy

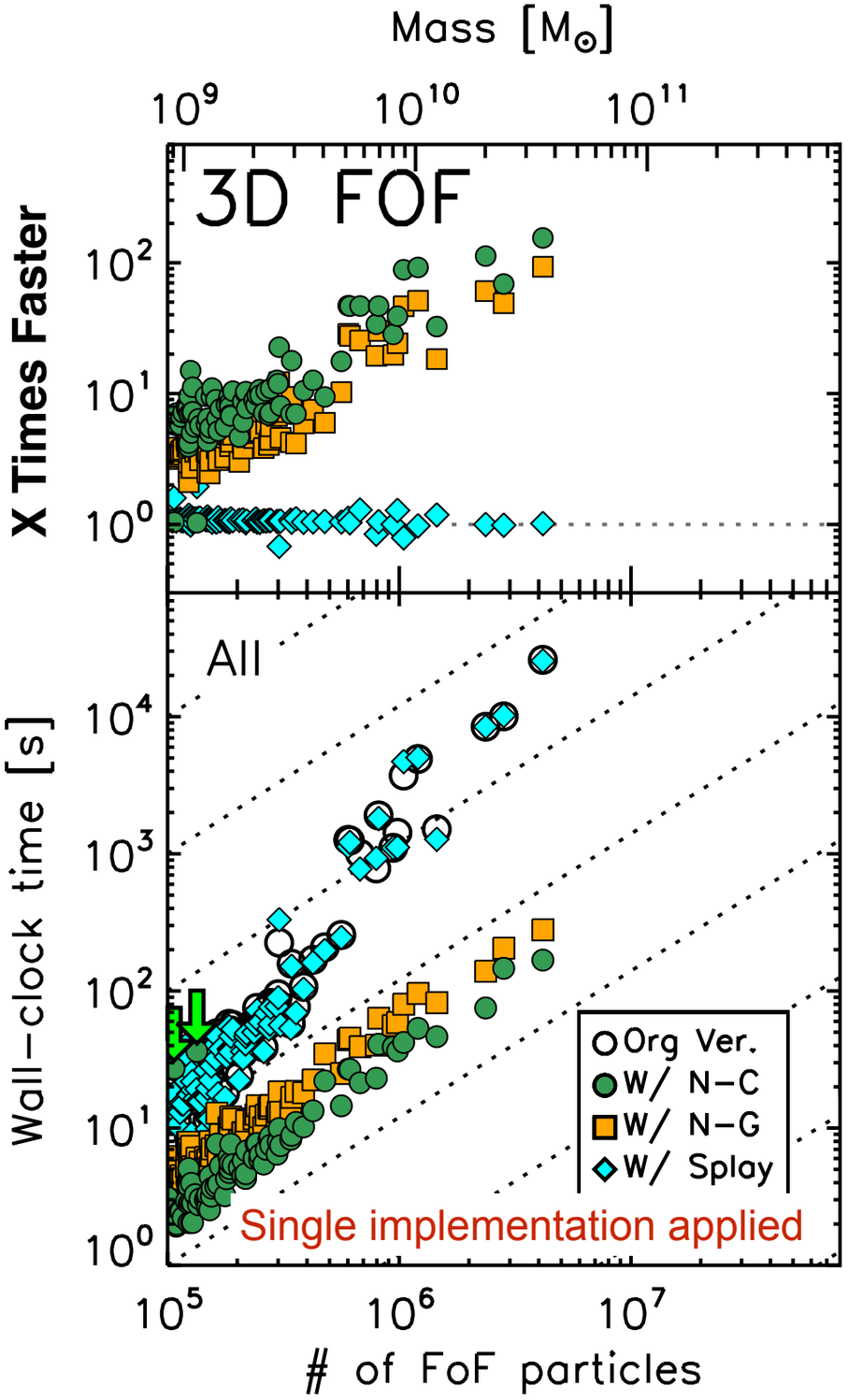
- Node-Closure
- Node-Geometry
- Splay

  Effectively find new member particles.

  But, do not affect the number of calculations.



3D FoF Results



Well operating for 3D FoF

- Node-Closure & Node-Geometry work well

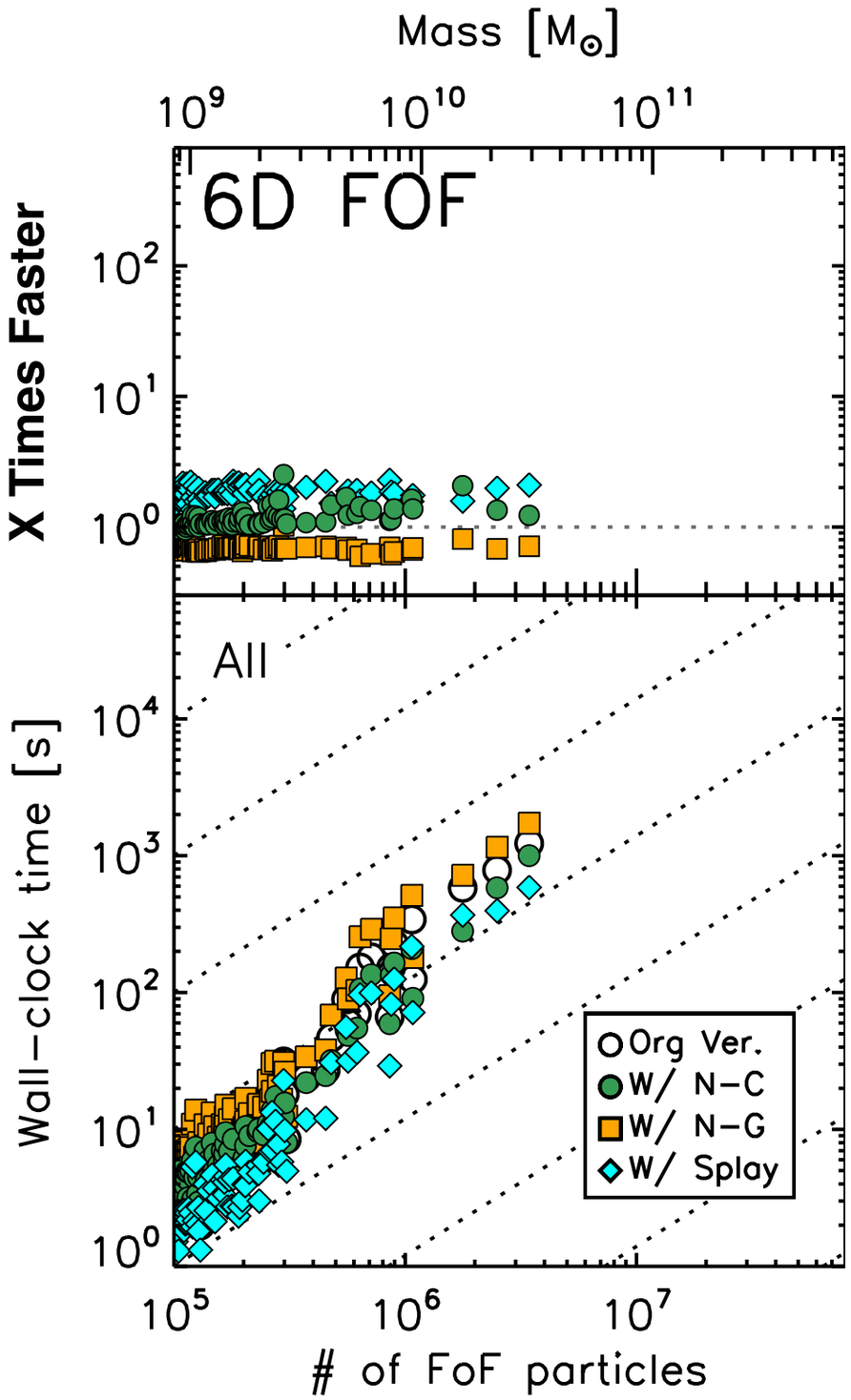
  Shallowing tree depth is enough

- Splay has no improvements

  Splay itself does not change the number of distance calculations

- Poor performance in 3D FoF is due to the repetitive visits to fully-connected nodes

6D FoF Results

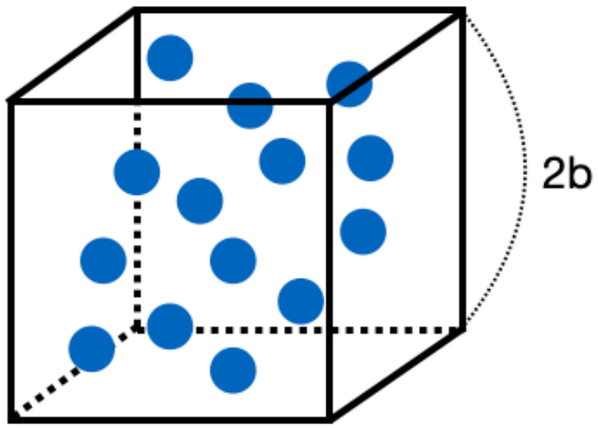


Not effective for 6D FoF

- All single implementations do not show significant improvement
- Shallowing the tree depth is ineffective
- Maybe due to the curse of dimensionality

Curse of Dimensionality

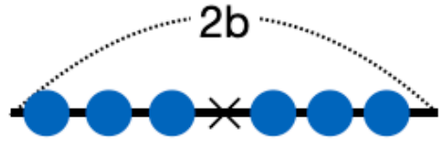
•Why 6D FoF is hard?



- ✳ N particles are in a box
- ✳ Uniform distribution
- ✳ Center as the origin (0,0,0)

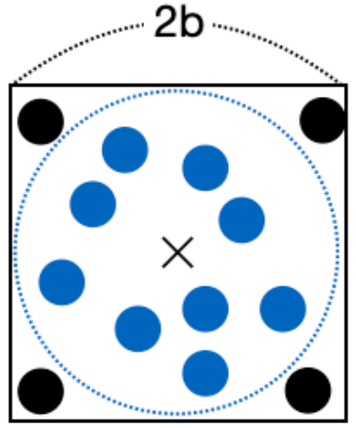
Suppose we want to find particles with which the distance from the center is less than b

1D Distance



All particles!

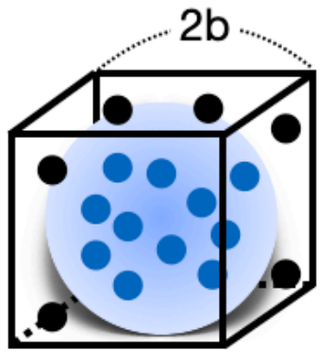
2D Distance



Most of them

The fraction is $\frac{\pi b^2}{(2b)^2}$

3D Distance

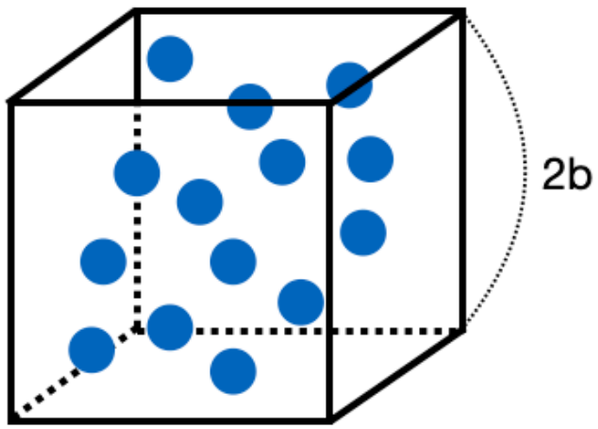


Some of them

The fraction is $\frac{4/3\pi b^3}{(2b)^3}$

Curse of Dimensionality

•Why 6D FoF is hard?



- ✳ N particles are in a box
- ✳ Uniform distribution
- ✳ Center as the origin (0,0,0)

Many visits are required for a node to be closed in high dimensions

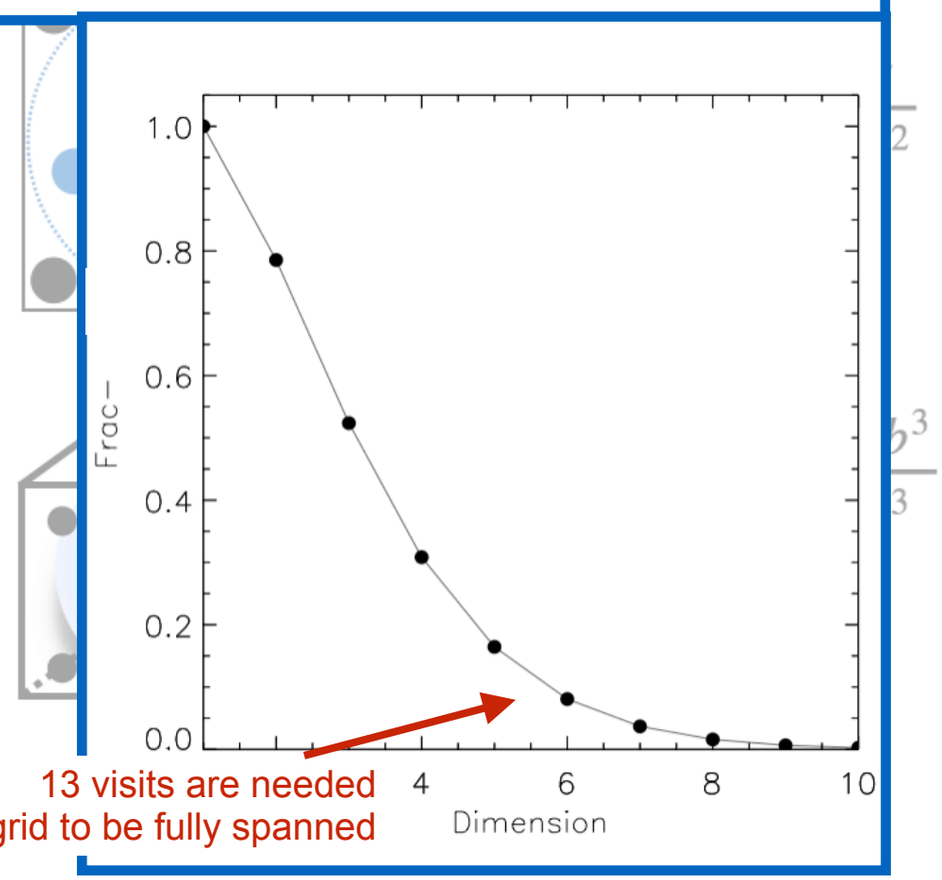
Generalization in N-dimension

$$\frac{\frac{\pi^{n/2}}{\Gamma(n/2+1)}b^n}{(2b)^n}$$

← Volume of n-dimensional sphere

← Volume of n-dimensional grid

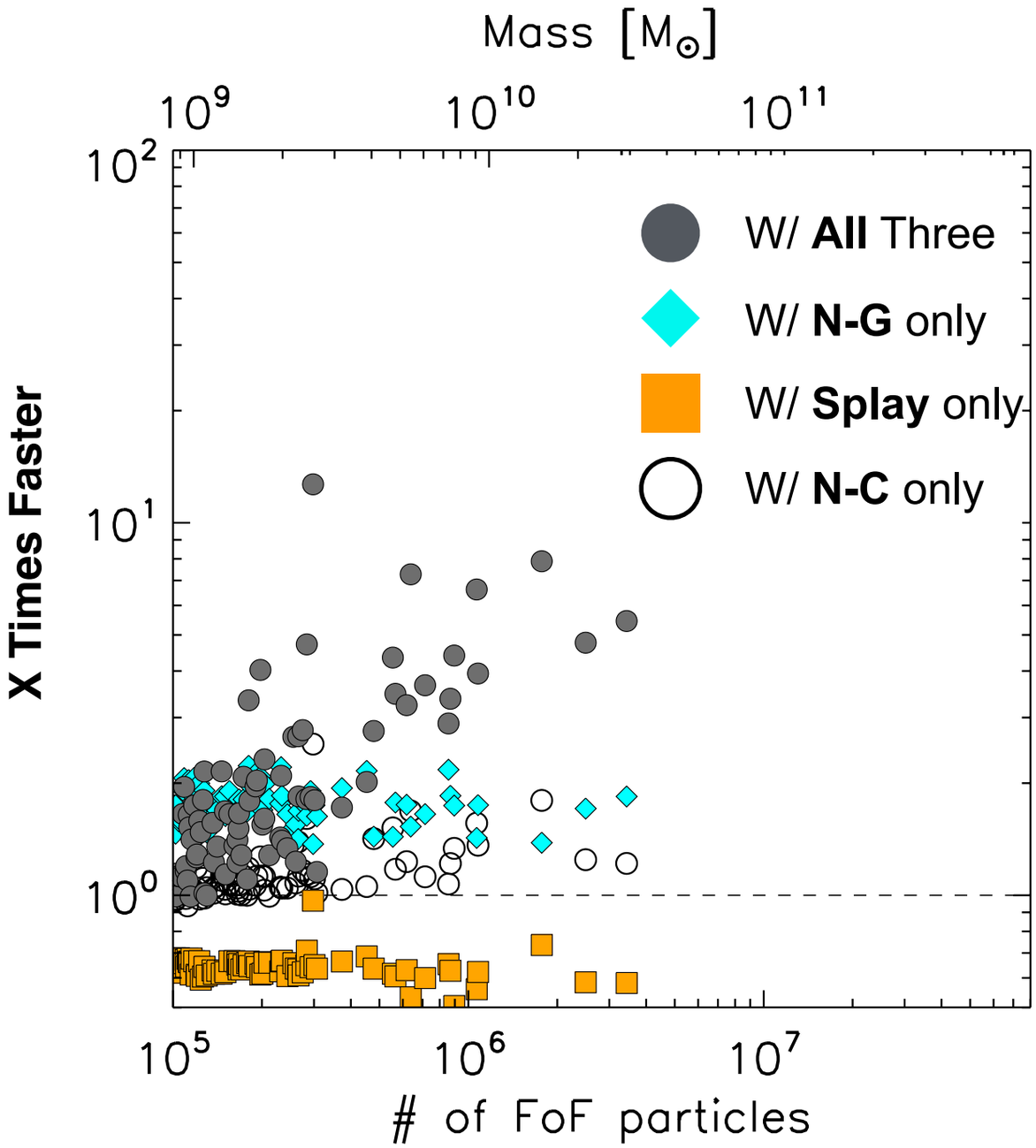
3D Distance



13 visits are needed
for the 6D grid to be fully spanned

Combining Implementations

Single Implementation is not enough.



When applying simultaneously,

- ▶ 8-10 times faster than before

  Partially solve the curse of dimensionality in 6D

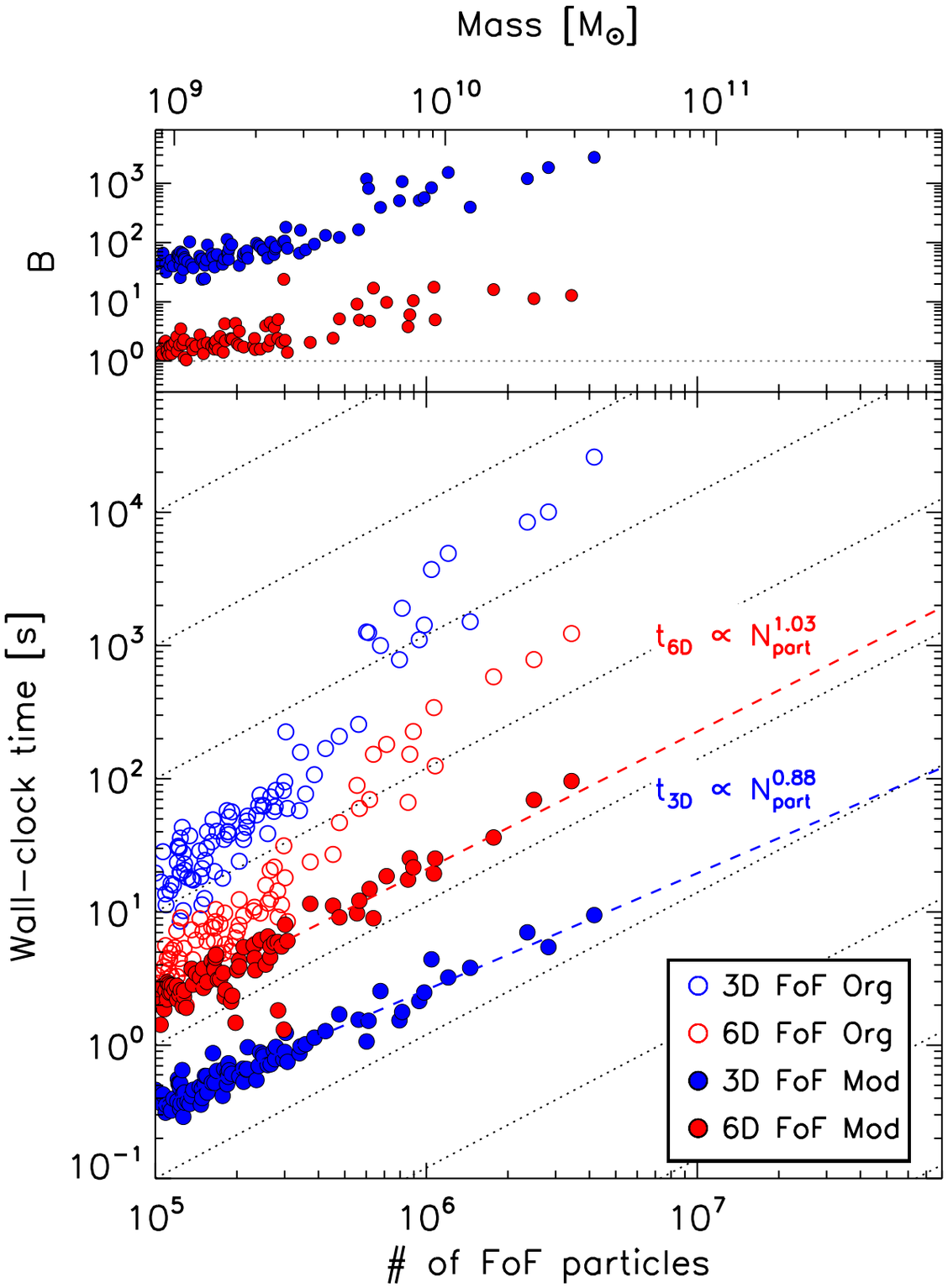
- ▶ Splay finds member particles effectively

  > Node get closed with fewer visits

  > Closed nodes are no longer opened

  by the N-C scheme

Updated Performance



Successfully improve the FoF efficiency

- Ideal scalings are obtained
- Finding for all consecutive NH snapshots (834)

  6-10 hours / snap (w/ 12 threads)
- Merger trees are built with TREEFrog (Elahi et al. 2019)
- Catalog is now available

Conclusion

- 6D FoF halo/galaxy finder will be the next generation of galaxy finder

  ✓ Due to its power of identifying substructures in high-dense regions

- Current (6D) FoF method shows very poor performance when applied to a high-resolution simulation

  ✓ Too many distance calculations are required in over-dense regions even with the aid of tree structures

- Performance enhancement of 6D FoF galaxy finder on a high-resolution simulations

  ✓ Achieve the ideal scaling of wall-clock times for identification

  ✓ Our implementations can be applied to any tree-based FoF codes